IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :	Taner, M. et al.	Art Unit :	2863
Serial No.:	10/035,955	Examiner :	Le, T.
		Docket No.:	RSI-003

Filed : 12/24/2001
Title : SYSTEM FOR UTILIZING SEISMIC DATA TO ESTIMATE SUBSURFACE LITHOLOGY

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF M. TURHAN TANER

M. Turhan Taner declares that:

I am currently employed by and have been employed by RDSP I, LP, the assignee of all right, title and interest in the referenced patent application since 1994 as a research geophysicist. I have published numerous papers on the subject of seismic attributes and their application to interpretation of seismic data. I am the same person who authored a publication cited in an Office Action dated November 6, 2003 in the referenced patent application entitled, *Kohonen's Self-Organizing Networks With "Conscience"*, Seismic Research Corporation. I have worked on various research projects related to Kohonen self-organizing maps since at least the time of publication of the foregoing publication.

During late 1999, and in the regular course of my employment with RDSP I, LP, I conceived of a way to calibrate self organizing map clusters for use in reservoir characterization. I worked on an number of experimental computer programs intended to embody the concept. A result of my development work is memorialized in a report for internal review at RDSP I, LP entitled, *Calibration of Self-Organizing Maps*, produced in November 2000. A copy of that report is attached as Exhibit A. The report expressly explains a calibration method for providing a relationship between each self organized map and wellbore-measured lithology.

Experimental computer source code intended to embody the calibration method described in the above report was generated as early as February 2000, and was revised to improve its performance in January 2001. A header comment table from the source code is attached as Exhibit B to show date of creation of the computer source code.

All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted,

M. Turhan Taner

EXHIBIT A

CALIBRATION OF SELF-ORGANIZED MAP CLUSTERS

M. Turhan Taner, Rock Solid Images **November, 2000**

INTRODUCTION

Kohonen's Self Organizing Feature Maps (SOFM) and other unsupervised clustering methods generate groups based on the identification of various discriminating features. These methods seek an organization in the dataset and form relational organized clusters. However, these clusters may or may not have any physical analogues in the real world. In order to relate these clusters to the real world, we have to develop some form of calibration method that not only defines the relationship between the clusters and real world physical properties, it should also provide us with an estimate of the validity of these relationships. With the development of a calibrated relationship, the whole dataset can be classified. The principal steps, therefore, are the Three-C's **"Clustering , Calibration and Classification".**
The clustering step reduces the multiple dimensions of the data description into logically smaller groups. Each original data point defined by multiple attributes is reduced to one or two-dimensional relational groups. This establishes some logical clustering and reduces the complexity of the classification problem. Furthermore, calibration should be more successful since it will have to consider less variability in the data.
In this paper, I propose a simple calibration method that employs Bayesian logic to provide the relationship between cluster centers and the real world. The output will give the most probable calibration between each Self-Organized Map node and the wellbore-measured lithology. The second part of the output will give the probability of the calibration.

METHOD

A Bayesian decision is based on the knowledge of the probability density function of each class. The decision boundary between classes is located at the point where the probability density of adjoining classes is equal.

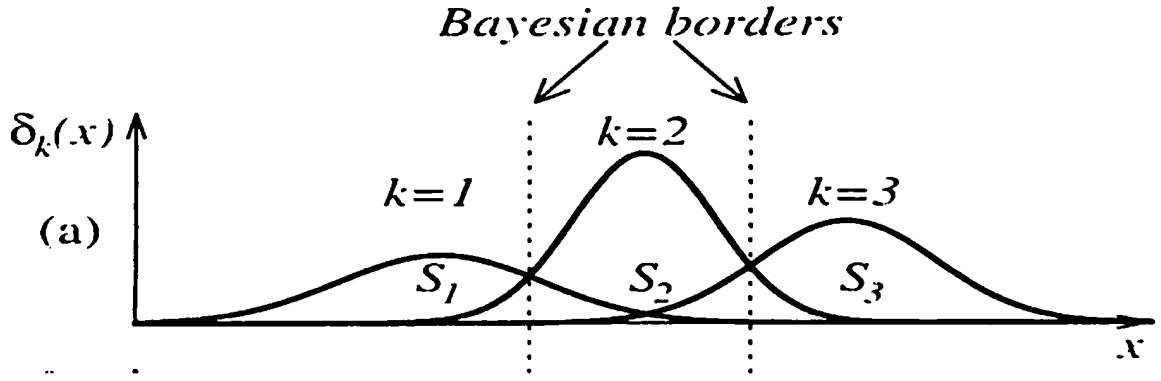


Figure 1. Bayesian Boundaries for Three Different Probability Densities

Figure 1 shows three different class probability densities. The Bayesian decision boundaries are located where the probabilities of different classes are equal. This is a very intuitive concept and easy to accept. Samples will be classified as belonging to the class with highest probability density.
In the method presented here, we will use Bayesian logic to establish the relationship between lithology classes and the SOM neural nodes. To establish such a relationship, we will need to compute the probability density function of each class in the SOM topology. I will use the Euclidean distance and the scaled Gaussian function as the probability density estimator.

Let ***w(i,j)*** represent the SOM ***i***'th weight of ***j***'th neuron and , ***X(i,n)*** represent the ***i***'th attributes of the ***n***'th lithology class. The Euclidean distance between the neural node and the input data sample is given by:

$$d(j,n) = \sqrt{\sum_{i=1}^{NI} \{ w(i,j) - X(i,n) \}^2} \quad , \tag{1}$$

where ***NI*** is the number of attributes (number of input dimensions).

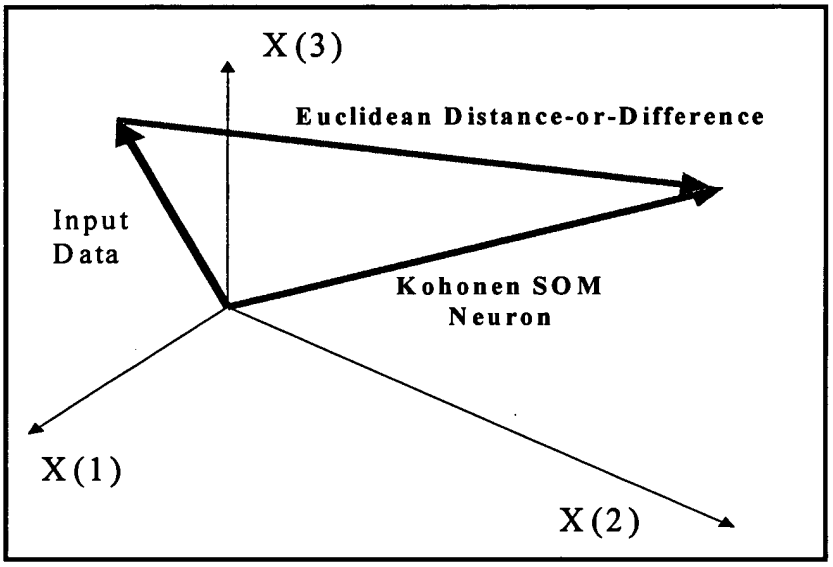


Figure 2 Input Vector, Neural Weight Vector and Euclidean Distance

During the SOM iteration, the Euclidean distances between data points and each neural node are computed. The node with the closest distance to the data is declared the winning neuron and its weights are adjusted to be closer to the input data. Its topologically neighboring neurons are also adjusted, but in a reduced amount which is proportional to their distance to the winning neuron. Iteration continues until an acceptable convergence is reached. Since the input data is not perfectly organized, we expect the clustering around each neuron to exhibit some scatter, i.e. some variance other than zero. In the calibration stage we need to determine the degree of convergence, so our probability estimate will have some basis. The average variance of clustering will give us a measure of the distance between SOM neuron cluster centers. This average will control the shape of the Gaussian function. I will use this as a control distance for 50% probability. I am considering that each data point is valid with some probability. It could belong to any one of the clusters of the SOM. However the probability of belonging to any group is a function of the distance to the neuron. Then the probability is computed as a Gaussian function of the distance.

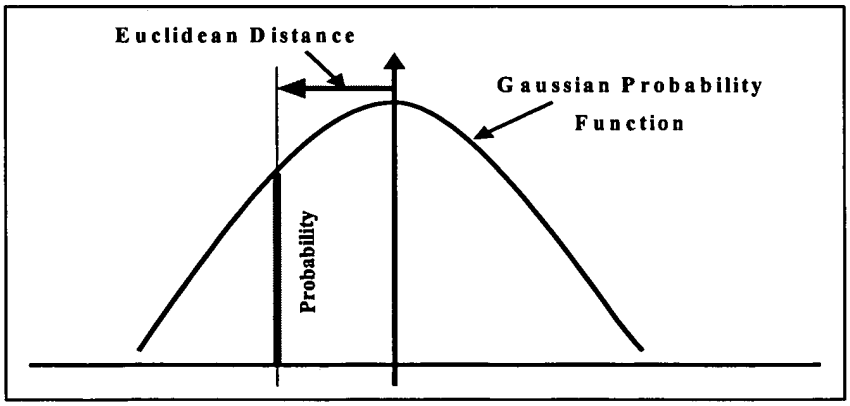


Figure 3 Computation of Probability by Euclidean Distance and Gaussian Function

In the first pass through the data we compute the average Euclidean distance between the input data samples and the winning neurons, which determines the a Gaussian shape factor α. In the second pass we use this shaping factor α in computation of the probability at each SOM neuron.

$$P(j,n) = exp\left(\alpha . d^2(j,n)\right), \qquad (2)$$

where $d(j,n)$ is the distance between n'th input data and j'th neuron.
This suggests that the closer the data point is to a node, the higher the probability of a correct calibration.
We generate a probability map, with the same topology as the SOM for each lithology class. For each lithology class data point we compute the distance and the (Gaussian function) probability for each SOM neural node. We accumulate these probabilities for data samples for that particular lithology class. Finally, we compute a scalar and divide the accumulated probabilities so the sum is equal to unity (100%). This map now represents the probability density of the particular lithology class.
We compare this lithology probability map with the maximum probability map (MPM) using Bayesian logic. We update the MPM if the lithology probability map contains a higher probability than the MPM and we also update the classification number, otherwise the MPM is left unaltered
This procedure is repeated for all classes. Upon completing the computation for all of the classes, we will have a table of classes with the highest probability for each SOM neural node and a table of corresponding probability densities.

Since the data is given in list form containing the attribute values and corresponding lithology class, calibration could be conducted on multi-well data and for both deviated and horizontal wells, where synthetics are difficult to generate.

Data Example:

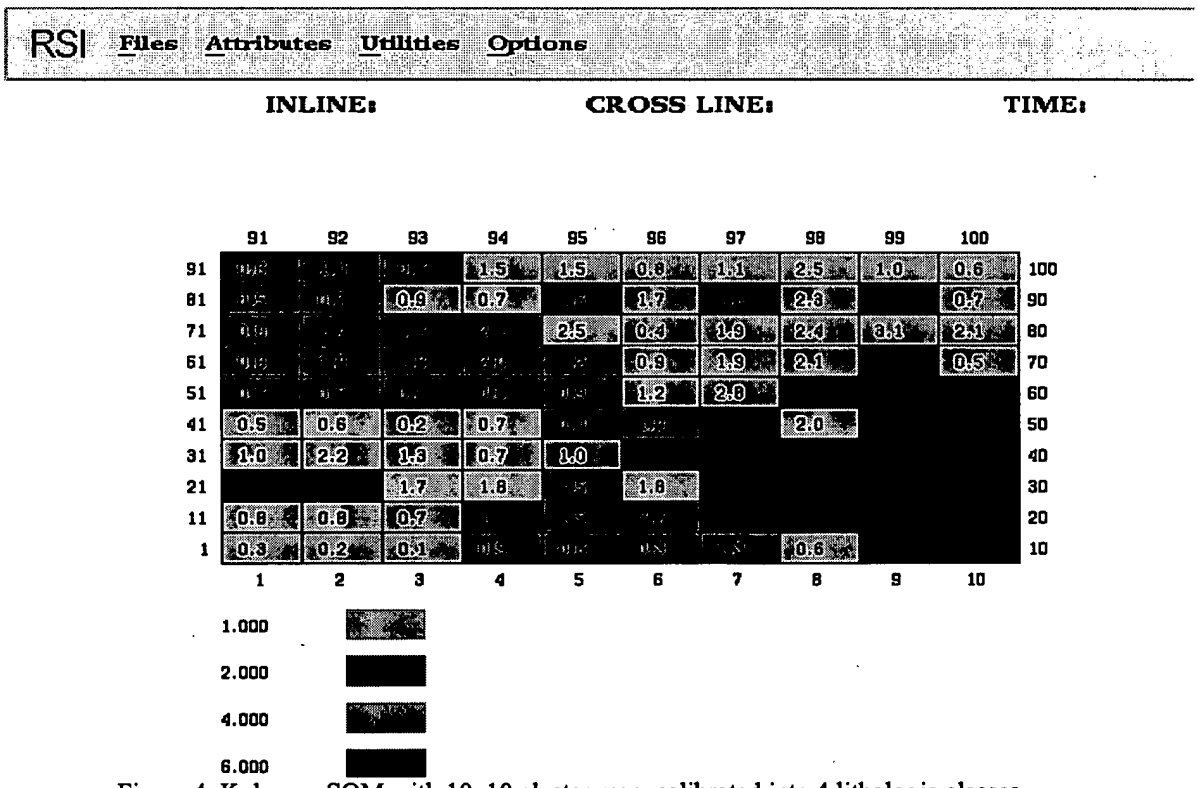


Figure 4: Kohonen SOM with 10x10 cluster map calibrated into 4 lithologic classes.
Each class is shown by a specific color for ease of recognition.
Number on each cluster center represent relative probability (Total adds up to 1.0)

Figure 4 shows an actual data calibration example (Dumas, 2000). User gave 4 different lithologies. Kohonen's SOM was run with 10x10 cluster topology. A probability field for each class was generated and scaled so the sum equal to 1.0 representing 100 percent probability. The probability function was generated using the RMS clustering distance as the 50 percent probability value. This gives us the value of Gaussian shaping factor. Smaller RMS values will make the Gaussian curve sharper, and larger values will produce a smoother curve. The maximum probability for each cluster center is determined by comparison. The program displays final calibration and related probabilities as shown on figure 4

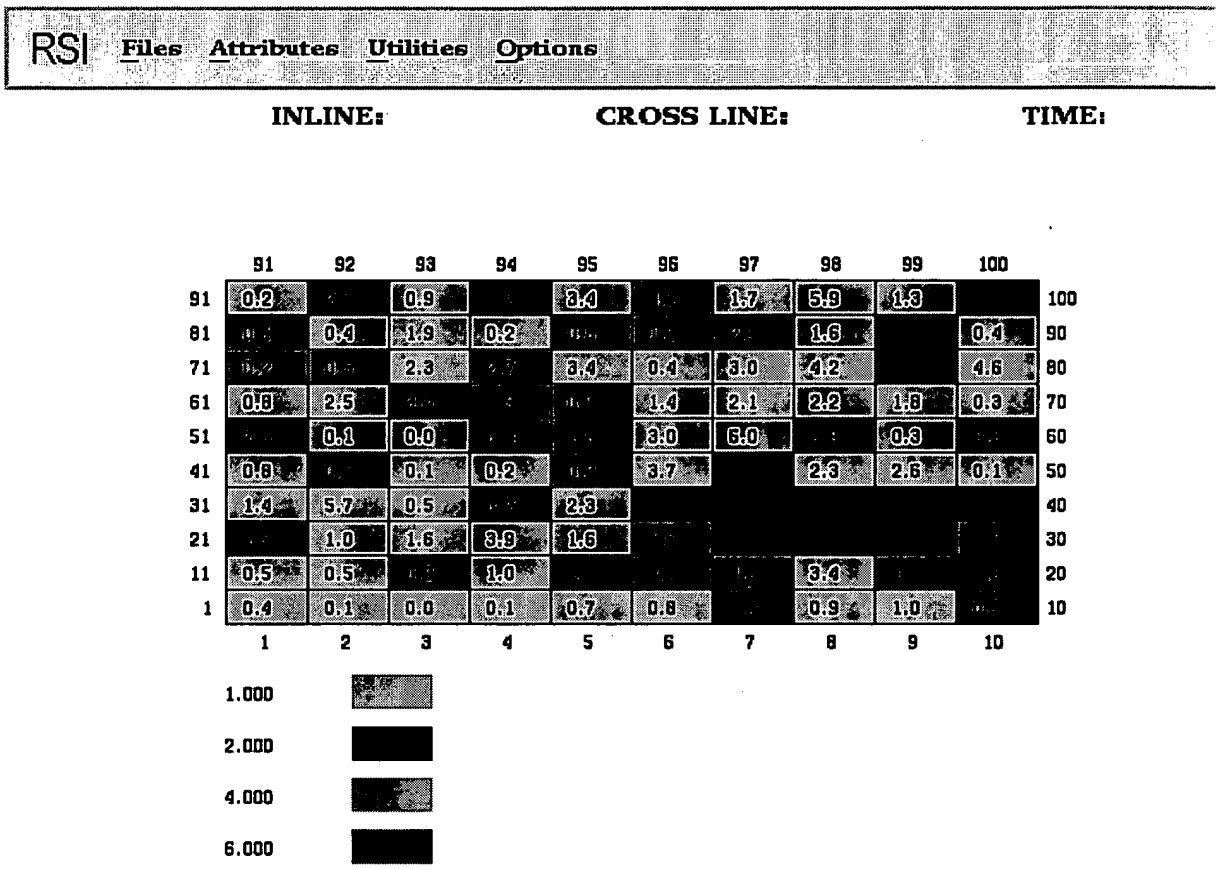


Figure 5 : SOM Calibration with Shaping factor computed from 1/2 RMS of clustering Error.

.In order to see the influence of the of the shaping factor , we used half of the RMS value, which will give a sharper probability curve. The corresponding results are shown on Figure 5.

Comparing the two figures, we observe on Figure 5 , as expected, more singular unconnected classes with higher probabilities of each calibration. This is due to the sharper Gaussian curve on Figure 5 than the one used on Figure 4. On additional comparison, we can see that the main calibration zones with higher probabilities remain about the same. This will give us additional confidence, that the calibration has reasonable probability.

We would expect calibration should appear over connected cluster centers. This is due to the nature of the mapping procedure, that the neighbors will have similar characteristics, proportional to their distance to each other. Therefore similar lithologies should appear adjoining around neighboring cluster centers. The figure 4 essentially shows such a clustering. It, however, has several singular calibration indicators. We will investigate the cause of this singularities and report in another paper. Initial thinking points to several reasons;

a) SOM may not have completely converged to a stable solution,

b) Lithology classification training set may not be uniquely defined, (a number of possible mis-directions),

c) High degree of uncertainty around singularities. In this case we may overrule the classification and decide to rule as the class of majority of the neighboring classes.

These hypotheses will have to be investigated by actual examples.

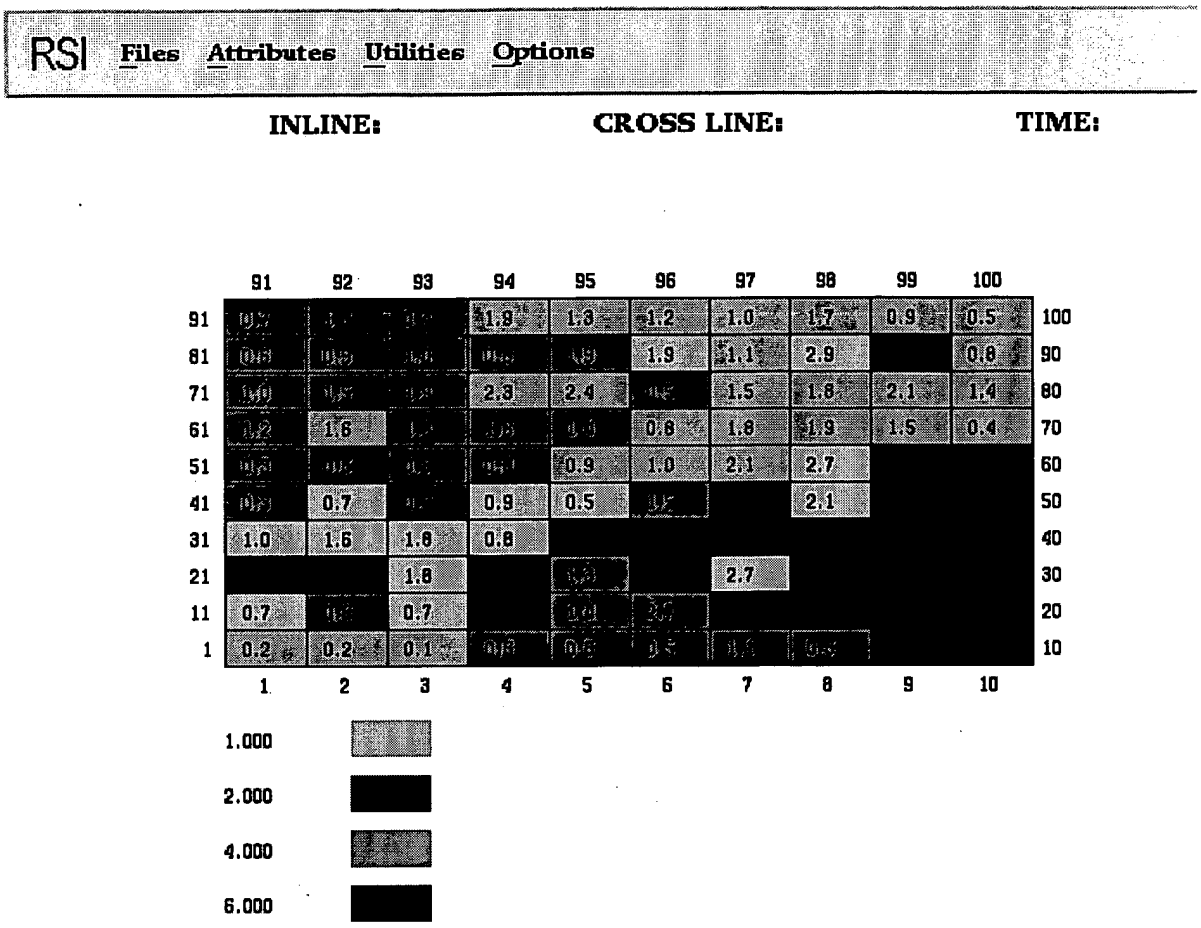


Figure 6 SOM Calibration with shaping factor same as Figure 1, with outliers suppressed.

One of the problem could be due to some noisy input or mis-classification due a various reasons, such as time tie, location error etc. These inputs will produce outliers in the averaging process during the probability density computation. Each input produce a probability value at each SOM cluster center between 0.0 and 1.0. The value of 1.0 means the perfect 100 percent probability. Therefore during the averaging stage, if the training data is well described, we should not expect to have variation from the mean, at the extreme, no more than 1.0. A reasonable estimate for outliers could be set at 0.5. That is if one of the input gives a probability 0.5 away from the mean, we will reduce its weight down to 110 percent. We will use , again, a Gaussian function to compute weights of all of the probabilities going into the sum according to their distance to the estimated mean value. We, then, compute a weighted mean. This will be used in several iteration, each time we update the mean and recompute the corresponding weights. Program at, this time, does 4 iterations.

The Figure 6 shown above is the result of computing the weighted mean as the probability density estimation. We used the RMS error value as the shaping factor of the Gaussian curve. Weights were computed with another Gaussian curve where the shaping factor produced 0.01 weight for difference from the mean at 0.5. This figure may be compared with 4, where averages were computed without any editing.

One of the additional functions is to create a "No-Calibration" class. This, of course, will be subjective. We will have to use some probability level as the limit, above which will be the acceptable level. All probabilities below this level will be marked as No-Calibration.

VERIFICATION OF RESULTS:

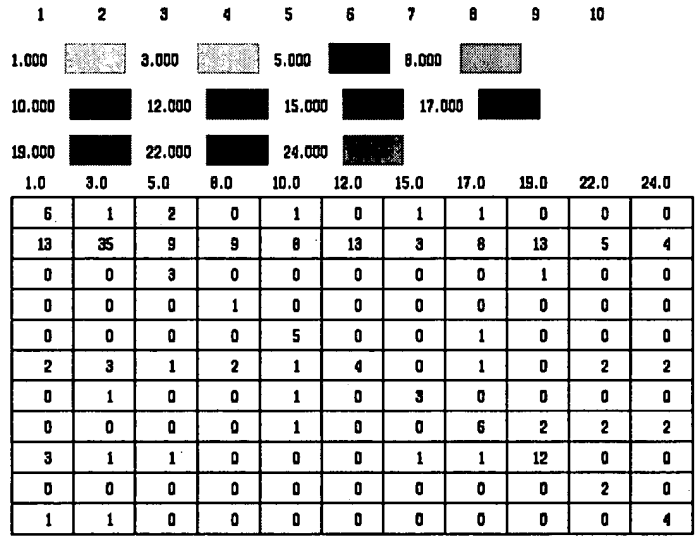
It is common for all neural network training, we will have to verify the results and see if the training data set is proper, or the network calibration is accurate. After the calibration is completed, we compute classification of the training data set by the calibrated network and check to see if predicted class matches the user given class. We have developed an error computation routine that shows error of the input data classes and the total number of errors on

each SOM neural node. In the acceptable conditions, we would expect to have random distribution of errors both on the data set and the neural nodes. Any accumulation of errors in a particular class or neural node may indicate a problem. The problem could be with the SOM computation or with the given training data set.

a) SOM problems could be due to insufficient convergence. This could be fixed by some additional computation time. SOM has two cycles,. The first one is organization using the conscience algorithm, that will try to produce equi-probable population densities. After a satisfactory grouping is reached, then the conscience algorithm is turned off and additional iterations are run to achieve convergence. The conscience algorithm is similar to the bias term in other neural networks. However, final clustering does not use this term. Therefore the last set of iteration must be run without the conscience algorithm.

The second possible problem could be due to the selected attributes may not be an effective set of discriminators. In this case, we have to experiment with the different sets of attributes, and select the ones with minimum number and randomly distributed errors.

b) The training data set errors could be detected if a particular class has most consistent errors. In this case we will have to review the data set and correct any possible lithology classification. The probabilistic nature of the program with outlier suppression logic can handle some erroneous classification in the input data set. However, if the erroneous input becomes the majority, then correction logic, based on the majority of data being correct, will not work. The errors could be due to incorrect location of the seismic attributes, in time and space, with respect to the actual well location. Checking synthetic versus actual data tie may help to alleviate the problem. In deviated or horizontal well cases several iterations of possible well locations may be necessary.



	1.0	3.0	5.0	8.0	10.0	12.0	15.0	17.0	19.0	22.0	24.0
1.0	6	1	2	0	1	0	1	1	0	0	0
3.0	13	35	9	9	8	13	3	8	13	5	4
5.0	0	0	3	0	0	0	0	0	1	0	0
8.0	0	0	0	1	0	0	0	0	0	0	0
10.0	0	0	0	0	5	0	0	1	0	0	0
12.0	2	3	1	2	1	4	0	1	0	2	2
15.0	0	1	0	0	1	0	3	0	0	0	0
17.0	0	0	0	0	1	0	0	6	2	2	2
19.0	3	1	1	0	0	0	1	1	12	0	0
22.0	0	0	0	0	0	0	0	0	0	2	0
24.0	1	1	0	0	0	0	0	0	0	0	4

Figure 7 Calibration Cross-Validation Table

The figure 7 shows cross-validation table of calibration run. Each row shows the classes given by the user. The number in each column indicate how many times that particular class was classified as the class corresponding to that column. If all classifications were correct, then all of the classifications would have been on the diagonal box. All classifications off-diagonal indicate possible error. In the example shown on figure 7 , Class 3 has considerable number of mis-classification. Others are basically have larger numbers on the diagonal. In this case we have problem with class 3. By comparing largest misclassifications, we may adjust the training data set until the mis-classification errors are reduced to some acceptable levels.

The LOG-ANN program also generates a text file indicating the SOM node numbers, classification, their probabilities, data errors and accumulated errors of each SOM node.

CONCLUSIONS

The calibration step connects the clustering and classification steps in highly logical manner. The procedure will perform a calibration for all SOM neurons regardless of the size and topology of the network.

As mentioned in the foregoing, any wellbore configuration can be accommodated; even those highly deviated cases where the generation of synthetic for calibration purposes may be difficult.
This procedure simplifies the Probabilistic Neural Network (PNN) approach. In the PNN procedure each training data point is considered a valid point in data space and a corresponding probability function is generated. in N-dimensional space. In our current implementation, we conduct the clustering at an SOM dimension where all attributes are well organized. This reduces the dimensionality of the problem and results in a considerable reduction in computation time. Since the data is clustered by the SOM, the calibration is less complicated, and most probably, more accurate.
The Three-C procedure is analogous to the regularized Radial Basis Function Networks (RBF). The original form of RBF uses each training data sample as the center of each neuron in the Hidden Layer. This will result in an enormous amount of hidden layer neurons. The regularization process reduces the number of neurons to a level that represents the input data field by a minimum number of neurons. Following this reduction, the output layer weights are computed to linearly interpolate the desired results. In our procedure we use SOM clustering in the first stage, which is similar to a regularization of the RBF network. The calibration stage is performed via Bayesian logic rather than linear interpolators.

References:

Duda, R. O. and Hart, P. E. , 1973, ***Pattern Classification and Scene Analysis;*** Wiley-Interscience Publication.

Dumas, D., 2000, ***LogAnn Program***, Rock Solid Images Technology.

Haykin, S. 1994, ***"Neural Networks, A Comprehensive Foundation"***, Published by Macmillan College Publishing Company, New York. (This book contains one of the most comprehensive discussions of all types of Neural Networks including the Self-Organized Maps).

Kohonen, T., 1988, ***"Self-Organizing and Associative Memory"***, 3rd. ed., Published by Springer Verlag, New York.

EXHIBIT B

```
C*********************************************************************
       SUBROUTINE  CALPROB ( IWELL, ATRIB, NOATR, NWELL, NOCLASS,
     *                       SOFM, NOXX, NOYY, MAPS,
     *                       CALIB, PROB,SDISA,SCALE,NITR,PAVE,WEG,
     *                       CLASS )
C
C*********************************************************************
**
C
      IMPLICIT NONE
      SAVE
C
C+CALIBRT
C
C-FUNCTION:   THIS SUBROUTINE CALIBRATES THE UNSUPERVISED CLUSTERING
C                    BASED ON GIVEN WELL BORE MEASUREMENTS
C                    BY GENERATING MAXIMUM PROBABILITY
C
C-CALLING SEQUENCE:
C
C           CALL CALPROB(WELL,SOFM,NWELL,NOXX,NOYY ....)
C
C-ARGUMENTS:
C
C  IWELL(*) = WELL BORE LITHOLOGY OR RESERVOIR CLASSIFICATION NUMBER
C  ATRIB(*) = ATTRIBUTES CORRESPONDING EACH WELL BORE SAMPLES
C  NOATR    = NUMBER OF ATTRIBUTE  SAME NUMBER AS SOM COEFFICIENTS)
C  NWELL    = NUMBER OF WELL BORE SAMPLES
C  NOCLASS  - TOTAL NUMBER OF WELL BORE CLASSES
C  SOFM(*)  = SELF ORGANIZING FEATURE MAP CLUSTER COEFFICIENTS
C             WELL BORE CLASSIFICATION NUMBER.
C  NOXX     = NUMBER OF NEURONS IN X DIRECTION.
C  NOYY     = NUMBER OF NEURONS IN Y DIRECTION
C  MAPS     = KOHONEN MAP TYPE. ( 1 = ONE D, 2 = RECTANGULAR, 3
=TRIANGULAR)
C
C----- OUTPUT ---------
C
C  CALIB(I) = WELL BORE LITHOLOGY CLASSES OF EACH INPUT SOM NEURON
C  PROB(I)  + PROBABILITY OF EACH CALIBRATION ( 0<PROB<100 )
C
C-DESCRIPTION:
C
C    BIG LOOP IS ON EACH LITHOLOGY CLASS;
C    FOR EACH LITHOLOGY CLASS, EACH DATA SAMPLE ( CONSISTING OF
ATTRIBUTES
C    PICKED FROM THE VICINITY OF WELLS) VECTOR DOT PRODUCT WILL BE
COMPUTED.
C    WITH EACH NEURON. THIS WILL BE THE PROBABILITY OF EACH DATA POINT.
C    THESE VALUES WILL BE ACCUMULATED FOR ALL DATA POINTS FOR THAT CLASS
AND
C    RESULTS WILL BE DIVIDED BY THE NUMBER OF POINTS. THIS WILL
CONSTITUTE
C    THE PROBABILITY OF THAT CLASS.
C    NEXT WE WILL USE BAYESIAN LOGIC, THAT IS COMPARE THIS PROBABILITY
FUNCTION
C    WITH THE STORED MAXIMUM PROBABILITY FUNCTION OF PREVIOUS
COMPUTATIONS.
C    FOR EACH NEURON, IF THE NEW ONE IS LESS THAN PREVIOUS ONE, THEN GO
TO THE
C    NEXT NEURON. IF GREATER , THEN UPDATE THE MAXIMUM PROBABILITY AND
```

```
SET THE
C    NEW CLASS NUMBER ON THE LIST FOR THAT NEURON.
C    REPEAT THIS FOR ALL THE CLASSES. AT THE END, WE WILL HAVE TWO
TABLE,
C    SIMILAR TO THE KOHONEN MAP; ONE CLASS ASSIGNMENT FOR EACH NEURON,
AND
C    THE SECOND ONE PROBABILITY OF THAT CLASS ASSIGNMENT. THESE TABLES
WILL
C    LATER BE USED FOR (CALIBRATED) CLASSIFICATION OF THE WHOLE DATA
VOLUME.
C
C-REVISED:  20-FEBRUARY-2000      BY   M. TURHAN TANER & NAUM DERZHI
C-REVISED:  16-JANUARY -2001      BY   M. TURHAN TANER
C
C
C++
C
```